(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号
特開2001－60947
（P2001－60947A）
(43)公開日　平成13年３月６日(2001.3.6)

(51)Int.Cl.[7]	識別記号	ＦＩ		ﾃｰﾏｺｰﾄﾞ(参考)
Ｈ０４Ｌ　9/32		Ｈ０４Ｌ	9/00	６７３Ａ
Ｇ０６Ｆ　15/00	３３０	Ｇ０６Ｆ	15/00	３３０Ｅ
Ｈ０４Ｌ　9/08		Ｈ０４Ｌ	9/00	６０１Ｃ
				６７５Ｂ

審査請求　未請求　請求項の数66　ＯＬ　（全 16 頁）

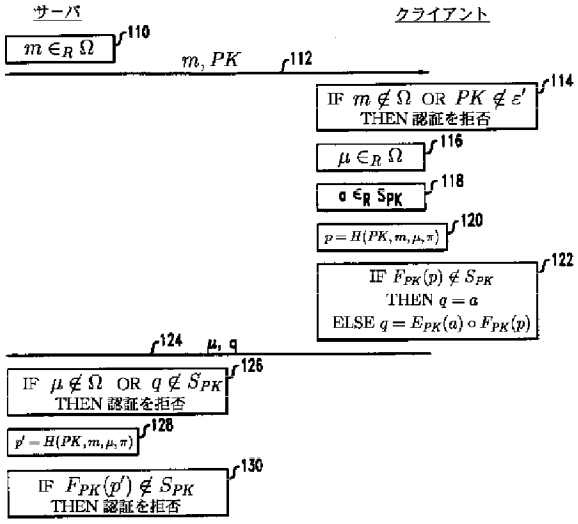
(21)出願番号　特願2000－210117(P2000－210117)
(22)出願日　平成12年７月11日(2000.7.11)
(31)優先権主張番号　０９／３５３４６８
(32)優先日　平成11年７月13日(1999.7.13)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　596077259
ルーセント　テクノロジーズ　インコーポレイテッド
Ｌｕｃｅｎｔ　Ｔｅｃｈｎｏｌｏｇｉｅｓ　Ｉｎｃ．
アメリカ合衆国　07974　ニュージャージー、マレーヒル、マウンテン　アベニュー　600－700
(74)代理人　100081053
弁理士　三俣　弘文

最終頁に続く

(54)【発明の名称】　相互ネットワーク認証方法

(57)【要約】
【課題】　安全性を証明可能なパスワード単独相互認証プロトコルを実現する。
【解決手段】　クライアントは、サーバから受信した公開鍵が、公開鍵暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定する。そのような要素でない場合、認証はクライアントによって拒否され、そうでない場合、プロトコルは続行される。一実施例では、クライアントとサーバはいずれも、認証目的で使用される１つのパスワードを所有する。クライアントは、少なくとも公開鍵およびパスワードの関数としてパラメータｐを生成する。公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が公開鍵のメッセージ空間の要素である場合、クライアントが、公開鍵を用いて公開鍵のメッセージ空間の実質的にランダムな要素を暗号化し、その結果と、F_{PK}（ｐ）との間で、公開鍵メッセージ空間の群演算を実行する。



【特許請求の範囲】

【請求項１】　公開鍵暗号方式を利用したクライアントとサーバの間の相互ネットワーク認証方法において、前記クライアントが、

前記サーバから公開鍵を受信するステップと、

前記公開鍵が、前記公開鍵暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップと、

前記公開鍵が前記テスト可能スーパーセットの要素でない場合、認証を拒否するステップとを有することを特徴とする相互ネットワーク認証方法。

【請求項２】　前記クライアントが、

前記公開鍵が前記テスト可能スーパーセットの要素である場合、

少なくとも前記公開鍵およびパスワードの関数としてパラメータｐを生成するステップと、

公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素でない場合、

パラメータｑを、前記公開鍵のメッセージ空間の実質的にランダムな要素とするステップと、

ｑを前記サーバに送信するステップとをさらに有することを特徴とする請求項１に記載の方法。

【請求項３】　ｐは、さらに、少なくとも、前記サーバから受信したパラメータｍの関数として生成されることを特徴とする請求項２に記載の方法。

【請求項４】　ｐは、さらに、少なくとも、実質的にランダムな数μの関数として生成されることを特徴とする請求項３に記載の方法。

【請求項５】　前記クライアントが、

前記公開鍵が前記テスト可能スーパーセットの要素である場合、

少なくとも前記公開鍵およびパスワードの関数としてパラメータｐを生成するステップと、

公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素である場合、

前記公開鍵を用いて前記公開鍵のメッセージ空間の実質的にランダムな要素を暗号化し、その結果とF_{PK}（ｐ）との間で、公開鍵メッセージ空間の群演算を実行することによって、パラメータｑを生成するステップと、

ｑを前記サーバに送信するステップとをさらに有することを特徴とする請求項１に記載の方法。

【請求項６】　ｐは、さらに、少なくとも、前記サーバから受信したパラメータｍの関数として生成されることを特徴とする請求項５に記載の方法。

【請求項７】　ｐは、さらに、少なくとも、実質的にランダムな数μの関数として生成されることを特徴とする請求項６に記載の方法。

【請求項８】　ｍおよびμをパラメータとしてディフィ・ヘルマンプロトコルを用いてセッション鍵を生成するステップをさらに有することを特徴とする請求項７に記載の方法。

【請求項９】　前記クライアントが、

前記公開鍵が前記テスト可能スーパーセットの要素である場合、

少なくとも前記公開鍵とパスワードの関数との関数としてパラメータｐを生成するステップと、

公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素でない場合、

パラメータｑを、前記公開鍵のメッセージ空間の実質的にランダムな要素とするステップと、

ｑを前記サーバに送信するステップとをさらに有することを特徴とする請求項１に記載の方法。

【請求項１０】　前記パスワードの関数は、前記パスワードの一方向性ハッシュ関数であることを特徴とする請求項９に記載の方法。

【請求項１１】　ｐは、さらに、少なくとも、前記サーバから受信したパラメータｍの関数として生成されることを特徴とする請求項９に記載の方法。

【請求項１２】　ｐは、さらに、少なくとも、実質的にランダムな数μの関数として生成されることを特徴とする請求項１１に記載の方法。

【請求項１３】　前記クライアントが、

前記公開鍵が前記テスト可能スーパーセットの要素である場合、

少なくとも前記公開鍵とパスワードの関数との関数としてパラメータｐを生成するステップと、

公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素である場合、

前記公開鍵を用いて前記公開鍵のメッセージ空間の実質的にランダムな要素を暗号化し、その結果と、F_{PK}（ｐ）との間で、公開鍵メッセージ空間の群演算を実行することによって、パラメータｑを生成するステップと、

ｑを前記サーバに送信するステップとをさらに有することを特徴とする請求項１に記載の方法。

【請求項１４】　前記パスワードの関数は、前記パスワードの一方向性ハッシュ関数であることを特徴とする請求項１３に記載の方法。

【請求項１５】　ｐは、さらに、少なくとも、前記サーバから受信したパラメータｍの関数として生成されることを特徴とする請求項１３に記載の方法。
【請求項１６】　ｐは、さらに、少なくとも、実質的にランダムな数μの関数として生成されることを特徴とする請求項１５に記載の方法。
【請求項１７】　ｍおよびμをパラメータとしてディフィ・ヘルマンプロトコルを用いてセッション鍵を生成するステップをさらに有することを特徴とする請求項１６に記載の方法。
【請求項１８】　公開鍵暗号方式を利用したクライアントとサーバの間の相互ネットワーク認証方法において、前記サーバが、
使用可能公開鍵暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素である公開鍵をクライアントに送信するステップと、
前記クライアントから、パラメータｑとして、前記公開鍵のメッセージ空間の要素を受信するステップとを有することを特徴とする相互ネットワーク認証方法。
【請求項１９】　前記クライアントで少なくとも前記公開鍵およびパスワードの関数として生成されたパラメータｐに、公開鍵空間マッピング関数F_{PK}を作用させた結果F_{PK}（ｐ）が、前記公開鍵のメッセージ空間の要素でない場合、
前記公開鍵のメッセージ空間の実質的にランダムな要素をパラメータｑとして受信するステップを有することを特徴とする請求項１８に記載の方法。
【請求項２０】　前記クライアントで少なくとも前記公開鍵およびパスワードの関数として生成されたパラメータｐに、公開鍵空間マッピング関数F_{PK}を作用させた結果F_{PK}（ｐ）が、前記公開鍵のメッセージ空間の要素である場合、
公開鍵メッセージ空間の実質的にランダムな要素を公開鍵暗号化した結果と、F_{PK}（ｐ）との間で、公開鍵メッセージ空間の群演算を実行した結果をパラメータｑとして受信するステップを有することを特徴とする請求項１８に記載の方法。
【請求項２１】　前記クライアントで少なくとも前記公開鍵とパスワードの関数との関数として生成されたパラメータｐに、公開鍵空間マッピング関数F_{PK}を作用させた結果F_{PK}（ｐ）が、前記公開鍵のメッセージ空間の要素でない場合、前記公開鍵のメッセージ空間の実質的にランダムな要素をパラメータｑとして受信するステップを有することを特徴とする請求項１８に記載の方法。
【請求項２２】　前記パスワードの関数は、前記パスワードの一方向性ハッシュ関数であることを特徴とする請求項２１に記載の方法。
【請求項２３】　前記クライアントで少なくとも前記公開鍵とパスワードの関数との関数として生成されたパラメータｐに、公開鍵空間マッピング関数F_{PK}を作用させた結果F_{PK}（ｐ）が、前記公開鍵のメッセージ空間の要素である場合、
公開鍵メッセージ空間の実質的にランダムな要素を公開鍵暗号化した結果と、F_{PK}（ｐ）との間で、公開鍵メッセージ空間の群演算を実行した結果をパラメータｑとして受信するステップを有することを特徴とする請求項１８に記載の方法。
【請求項２４】　前記パスワードの関数は、前記パスワードの一方向性ハッシュ関数であることを特徴とする請求項２３に記載の方法。
【請求項２５】　前記公開鍵暗号方式はＲＳＡであり、前記公開鍵はパラメータＮおよびｅからなり、
前記公開鍵は、前記サーバによって、
Ｎはある値より大きく、
ｅはＮより大きく、
ｅは素数であるように選択されることを特徴とする請求項１８に記載の方法。
【請求項２６】　前記公開鍵暗号方式はＲＳＡであり、前記公開鍵はパラメータＮおよびｅからなり、
前記公開鍵は、前記サーバによって、
Ｎはある値範囲内にあり、
ｅはある値範囲内にあり、
ｅは素数であるように選択されることを特徴とする請求項１８に記載の方法。
【請求項２７】　前記公開鍵暗号方式はＲＳＡであり、前記公開鍵はパラメータＮおよびｅからなり、
前記公開鍵は、前記サーバによって、
ｅは所定の値であり、
Ｎはある値範囲内にあるように選択されることを特徴とする請求項１８に記載の方法。
【請求項２８】　ＲＳＡ暗号方式を利用したクライアントとサーバの間の相互ネットワーク認証方法において、前記クライアントが、
前記サーバからＲＳＡ公開鍵（Ｎ，ｅ）を受信するステップと、
前記ＲＳＡ公開鍵（Ｎ，ｅ）が、前記ＲＳＡ暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップと、
前記ＲＳＡ公開鍵（Ｎ，ｅ）が前記テスト可能スーパーセットの要素でない場合、認証を拒否するステップとを有することを特徴とする相互ネットワーク認証方法。
【請求項２９】　前記ＲＳＡ公開鍵（Ｎ，ｅ）が、前記ＲＳＡ暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップは、前記クライアントが、
Ｎはある値より大きく、
ｅはＮより大きく、
ｅは素数であるかどうかを決定するステップを含むことを特徴とする請求項２８に記載の方法。
【請求項３０】　前記ＲＳＡ公開鍵（Ｎ，ｅ）が、前記

ＲＳＡ暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップは、前記クライアントが、
Ｎはある値範囲内にあり、
ｅはある値範囲内にあり、
ｅは素数であるかどうかを決定するステップを含むことを特徴とする請求項２８に記載の方法。
【請求項３１】　前記ＲＳＡ公開鍵（Ｎ，ｅ）が、前記ＲＳＡ暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップは、前記クライアントが、
ｅは所定の値であり、
Ｎはある値範囲内にあるかどうかを決定するステップを含むことを特徴とする請求項２８に記載の方法。
【請求項３２】　前記クライアントが、
前記ＲＳＡ公開鍵（Ｎ，ｅ）が前記ＲＳＡ暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素である場合、
少なくとも前記ＲＳＡ公開鍵（Ｎ，ｅ）およびパスワードの関数としてパラメータｐを生成するステップと、
公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素でない場合、
パラメータｑを、前記ＲＳＡ公開鍵のメッセージ空間の実質的にランダムな要素とするステップと、
ｑを前記サーバに送信するステップとをさらに有することを特徴とする請求項２８に記載の方法。
【請求項３３】　前記F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素であるかどうかを決定するステップは、
ｐとＮの最大公約数が１に等しいかどうかを決定するステップを含むことを特徴とする請求項３２に記載の方法。
【請求項３４】　ｐは、さらに、少なくとも、前記サーバから受信したパラメータｍの関数として生成されることを特徴とする請求項３２に記載の方法。
【請求項３５】　ｐは、さらに、少なくとも、実質的にランダムな数μの関数として生成されることを特徴とする請求項３４に記載の方法。
【請求項３６】　前記クライアントが、
前記ＲＳＡ公開鍵（Ｎ，ｅ）が前記ＲＳＡ暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素である場合、
少なくとも前記ＲＳＡ公開鍵（Ｎ，ｅ）およびパスワードの関数としてパラメータｐを生成するステップと、
公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素である場合、
ａを、前記ＲＳＡ公開鍵のメッセージ空間の実質的にランダムな要素として、ｑ＝（ｐ・a^e）ｍｏｄ　Ｎによりパラメータｑを生成するステップと、
ｑを前記サーバに送信するステップとをさらに有することを特徴とする請求項２８に記載の方法。
【請求項３７】　前記F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素であるかどうかを決定するステップは、
ｐとＮの最大公約数が１に等しいかどうかを決定するステップを含むことを特徴とする請求項３６に記載の方法。
【請求項３８】　ｐは、さらに、少なくとも、前記サーバから受信したパラメータｍの関数として生成されることを特徴とする請求項３６に記載の方法。
【請求項３９】　ｐは、さらに、少なくとも、実質的にランダムな数μの関数として生成されることを特徴とする請求項３８に記載の方法。
【請求項４０】　ｍおよびμをパラメータとしてディフィ・ヘルマンプロトコルを用いてセッション鍵を生成するステップをさらに有することを特徴とする請求項３９に記載の方法。
【請求項４１】　前記クライアントが、
前記ＲＳＡ公開鍵（Ｎ，ｅ）が前記ＲＳＡ暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素である場合、
少なくとも前記ＲＳＡ公開鍵（Ｎ，ｅ）とパスワードの関数との関数としてパラメータｐを生成するステップと、
公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素でない場合、
パラメータｑを、前記ＲＳＡ公開鍵のメッセージ空間の実質的にランダムな要素とするステップと、
ｑを前記サーバに送信するステップとをさらに有することを特徴とする請求項２８に記載の方法。
【請求項４２】　前記パスワードの関数は、前記パスワードの一方向性ハッシュ関数であることを特徴とする請求項４１に記載の方法。
【請求項４３】　前記F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素であるかどうかを決定するステップは、
ｐとＮの最大公約数が１に等しいかどうかを決定するステップを含むことを特徴とする請求項４１に記載の方法。
【請求項４４】　ｐは、さらに、少なくとも、前記サーバから受信したパラメータｍの関数として生成されるこ

とを特徴とする請求項４１に記載の方法。
【請求項４５】　ｐは、さらに、少なくとも、実質的にランダムな数μの関数として生成されることを特徴とする請求項４４に記載の方法。
【請求項４６】　前記クライアントが、
前記ＲＳＡ公開鍵（Ｎ，ｅ）が前記ＲＳＡ暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素である場合、
少なくとも前記ＲＳＡ公開鍵（Ｎ，ｅ）とパスワードの関数との関数としてパラメータｐを生成するステップと、
公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素である場合、
ａを、前記ＲＳＡ公開鍵のメッセージ空間の実質的にランダムな要素として、ｑ＝（ｐ・a^{e}）ｍｏｄ　Ｎによりパラメータｑを生成するステップと、
ｑを前記サーバに送信するステップとをさらに有することを特徴とする請求項２８に記載の方法。
【請求項４７】　前記パスワードの関数は、前記パスワードの一方向性ハッシュ関数であることを特徴とする請求項４６に記載の方法。
【請求項４８】　前記F_{PK}（ｐ）が前記ＲＳＡ公開鍵のメッセージ空間の要素であるかどうかを決定するステップは、
ｐとＮの最大公約数が１に等しいかどうかを決定するステップを含むことを特徴とする請求項４６に記載の方法。
【請求項４９】　ｐは、さらに、少なくとも、前記サーバから受信したパラメータｍの関数として生成されることを特徴とする請求項４６に記載の方法。
【請求項５０】　ｐは、さらに、少なくとも、実質的にランダムな数μの関数として生成されることを特徴とする請求項４９に記載の方法。
【請求項５１】　ｍおよびμをパラメータとしてディフィ・ヘルマンプロトコルを用いてセッション鍵を生成するステップをさらに有することを特徴とする請求項５０に記載の方法。
【請求項５２】　公開鍵暗号方式を利用したクライアントとサーバの間の相互認証方法において、
前記サーバが、公開鍵を前記クライアントに送信するステップと、
前記クライアントが、前記公開鍵が前記公開鍵暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップと、
前記クライアントが、前記公開鍵が前記テスト可能スーパーセットの要素でないと決定した場合、前記クライアントが、認証を拒否するステップとを有することを特徴とする相互認証方法。
【請求項５３】　前記クライアントが、前記公開鍵が前記テスト可能スーパーセットの要素であると決定した場合、
前記クライアントが、少なくとも前記公開鍵およびパスワードの関数としてパラメータｐを生成するステップと、
前記クライアントが、公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップとをさらに有することを特徴とする請求項５２に記載の方法。
【請求項５４】　前記クライアントが、F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素であると決定した場合、
前記クライアントが、前記公開鍵を用いて前記公開鍵のメッセージ空間の実質的にランダムな要素ａを暗号化し、その結果とF_{PK}（ｐ）との間で、公開鍵メッセージ空間の群演算を実行することによって、パラメータｑを生成するステップと、
ｑを前記サーバに送信するステップとをさらに有することを特徴とする請求項５３に記載の方法。
【請求項５５】　前記クライアントが、F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素でないと決定した場合、
前記クライアントが、パラメータｑを、前記公開鍵のメッセージ空間の実質的にランダムな要素とするステップをさらに有することを特徴とする請求項５４に記載の方法。
【請求項５６】　前記サーバが、ｑが前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
前記サーバが、ｑが前記公開鍵のメッセージ空間の要素でないと決定した場合、前記サーバが、認証を拒否するステップと、
前記サーバが、ｑが前記公開鍵のメッセージ空間の要素であると決定した場合、前記サーバが、前記公開鍵および前記パスワードの関数としてパラメータｐ′を生成するステップと、
前記サーバが、公開鍵空間マッピング関数F_{PK}をｐ′に作用させた結果F_{PK}（ｐ′）が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
前記サーバが、F_{PK}（ｐ′）が前記公開鍵のメッセージ空間の要素でないと決定した場合、前記サーバが、認証を拒否するステップと、
前記サーバが、F_{PK}（ｐ′）が前記公開鍵のメッセージ空間の要素であると決定した場合、前記サーバが、
ｑとF_{PK}（ｐ′）との間で、公開鍵メッセージ空間の群演算の逆を実行し、その結果を、前記公開鍵に対応する秘密鍵を用いて復号することによって、パラメータａ′を生成するステップと、

ｒ＝ｈ（ａ′）を生成するステップと、
ｒを前記クライアントに送信するステップとをさらに有することを特徴とする請求項５４に記載の方法。
【請求項５７】　前記クライアントが、
ａ）F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素であるかどうか、および、
ｂ）ｒ＝ｈ（ａ）であるかどうかを決定するステップと、
ａ）またはｂ）が真でない場合、前記クライアントが、認証を拒否するステップと、
ａ）およびｂ）が真である場合、前記クライアントが、
ｔ＝ｈ′（ａ）を生成するステップと、
ｔを前記サーバに送信するステップとをさらに有することを特徴とする請求項５６に記載の方法。
【請求項５８】　前記サーバが、ｔ＝ｈ′（ａ′）であるかどうかを決定するステップと、
ｔ＝ｈ′（ａ′）である場合、前記サーバが、認証を受容するステップと、
ｔ≠ｈ′（ａ′）である場合、前記サーバが、認証を拒否するステップとをさらに有することを特徴とする請求項５７に記載の方法。
【請求項５９】　前記サーバおよび前記クライアントが認証を受容した場合、前記サーバおよび前記クライアントがが、後の安全な通信のためのセッション鍵を計算するステップをさらに有することを特徴とする請求項５８に記載の方法。
【請求項６０】　前記クライアントが、前記公開鍵が前記テスト可能スーパーセットの要素であると決定した場合、
前記クライアントが、少なくとも前記公開鍵とパスワードの関数との関数としてパラメータｐを生成するステップと、
前記クライアントが、公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップとをさらに有することを特徴とする請求項５２に記載の方法。
【請求項６１】　前記クライアントが、F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素であると決定した場合、
前記クライアントが、前記公開鍵を用いて前記公開鍵のメッセージ空間の実質的にランダムな要素ａを暗号化し、その結果とF_{PK}（ｐ）との間で、公開鍵メッセージ空間の群演算を実行することによって、パラメータｑを生成するステップと、
ｑを前記サーバに送信するステップとをさらに有することを特徴とする請求項６０に記載の方法。
【請求項６２】　前記クライアントが、F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素でないと決定した場合、
前記クライアントが、パラメータｑを、前記公開鍵のメッセージ空間の実質的にランダムな要素とするステップをさらに有することを特徴とする請求項６１に記載の方法。
【請求項６３】　前記サーバが、ｑが前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
前記サーバが、ｑが前記公開鍵のメッセージ空間の要素でないと決定した場合、前記サーバが、認証を拒否するステップと、
前記サーバが、ｑが前記公開鍵のメッセージ空間の要素であると決定した場合、前記サーバが、前記公開鍵と前記パスワードの関数との関数としてパラメータｐ′を生成するステップと、
前記サーバが、公開鍵空間マッピング関数F_{PK}をｐ′に作用させた結果F_{PK}（ｐ′）が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
前記サーバが、F_{PK}（ｐ′）が前記公開鍵のメッセージ空間の要素でないと決定した場合、前記サーバが、認証を拒否するステップと、
前記サーバが、F_{PK}（ｐ′）が前記公開鍵のメッセージ空間の要素であると決定した場合、前記サーバが、
ｑとF_{PK}（ｐ′）との間で、公開鍵メッセージ空間の群演算の逆を実行し、その結果を、前記公開鍵に対応する秘密鍵を用いて復号することによって、パラメータａ′を生成するステップと、
ｒ＝ｈ（ａ′）を生成するステップと、
ｒを前記クライアントに送信するステップとをさらに有することを特徴とする請求項６１に記載の方法。
【請求項６４】　前記クライアントが、
ａ）F_{PK}（ｐ）が前記公開鍵のメッセージ空間の要素であるかどうか、および、
ｂ）ｒ＝ｈ（ａ）であるかどうかを決定するステップと、
ａ）またはｂ）が真でない場合、前記クライアントが、認証を拒否するステップと、
ａ）およびｂ）が真である場合、前記クライアントが、
ｔ＝ｈ′（ａ）を生成するステップと、
ｔを前記サーバに送信するステップとをさらに有することを特徴とする請求項６３に記載の方法。
【請求項６５】　前記サーバが、ｔ＝ｈ′（ａ′）であるかどうかを決定するステップと、
ｔ＝ｈ′（ａ′）である場合、前記サーバが、認証を受容するステップと、
ｔ≠ｈ′（ａ′）である場合、前記サーバが、認証を拒否するステップとをさらに有することを特徴とする請求項６４に記載の方法。
【請求項６６】　前記サーバおよび前記クライアントが認証を受容した場合、前記サーバおよび前記クライアントがが、後の安全な通信のためのセッション鍵を計算するステップをさらに有することを特徴とする請求項６５

に記載の方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、ネットワーク認証および鍵交換に関し、特に、パスワード単独で(password-only)安全な相互ネットワーク認証および鍵交換プロトコルに関する。
【０００２】
【従来の技術】ネットワークを通じての認証は、リモートクライアントがネットワークサーバにアクセスすることを可能にするシステムの安全性（セキュリティ）の重要部分である。認証は一般に、次のうちの１つ以上を確認することによって実行される。
・ユーザが知っているもの（例えばパスワード）
・ユーザ自体、すなわち、指紋のような生物測定学的情報
・ユーザが持っているもの、すなわち、スマートカードのような何らかの識別トークン
例えば、現金自動預入支払機（ＡＴＭ）は、これらのうちの２つ、すなわち、ユーザが持っているもの（ＡＴＭカード）と、ユーザが知っているもの（個人識別番号（ＰＩＮ：personal identification number）とを確認する。ＡＴＭ認証は、データネットワークを通じての認証よりもずっと簡単である。その理由は、ＡＴＭ自体が信頼されたハードウェアであるとみなされており、ＡＴＭカードの存在を確認し、正しい情報を安全に中央取引サーバに転送することが信頼されているからである。
【０００３】認証に加えて、鍵交換は、データネットワークを通じての通信の重要部分である。クライアントとサーバが認証された後、安全な通信チャネルをそれらの間に設定しなければならない。これは一般に、クライアントとサーバが、認証の後の通信中に使用するための鍵を交換することによって実行される。
【０００４】データネットワーク、特に、インターネットのような公衆データネットワークを通じての認証は困難である。その理由は、クライアントとサーバの間の通信が多くの異なるタイプの攻撃を受けやすいからである。例えば、盗聴攻撃では、敵は、クライアントとサーバの間の通信を傍受することによって秘密の情報を知ることがある。敵がパスワード情報を知った場合、敵は、その情報を再生（リプレイ）してサーバに送って真正なクライアントになりすますことが可能となる（リプレイ攻撃(replay attack)という）。リプレイ攻撃は、クライアントから送られたパスワードが暗号化されている場合でも有効である。その理由は、敵は、実際のパスワードを知る必要がなく、代わりに、サーバが真正なクライアントから期待するもの（この場合は、暗号化されたパスワード）をサーバに提供すればよいからである。もう１つのタイプの攻撃にスプーフィング(spoofing)攻撃がある。この攻撃では、敵がサーバになりすまし、クライアントは、真正なサーバと通信していると信じるが、実際は、そうではなく敵と通信しているというものである。このような攻撃では、クライアントは敵に機密情報を提供してしまうことがある。
【０００５】さらに、パスワードに基づく認証プロトコルでは、パスワードが弱く、辞書攻撃を受けやすい可能性がある。辞書攻撃とは、目的のパスワードに関するいくつかの既知情報に対して、多数の可能性のあるパスワード（例えば、英語辞書内の全単語）をテスト（検査）することによって実行される、パスワードに対する力づくの攻撃である。既知情報は、例えば、公に入手可能な情報や、上記の方法のうちの１つにより敵によって取得された情報である。ユーザはしばしば、容易に覚えられ、容易に推測されるパスワードを選択するため、辞書攻撃はしばしば有効である。
【０００６】ネットワーク認証にはさまざまな技術が知られている。これらの知られている技術は、２つの分類に分けることができる。第１の分類は、クライアントシステム上に永続的に記憶されたデータを必要とする技術からなる。第２の分類は、クライアントシステム上に永続的に記憶されたデータを必要としない技術からなる。
【０００７】第１の分類に関して、永続的に記憶されたデータには、決して暴露されてはならない秘密データ（例えば、認証サーバと共有する秘密鍵）や、秘密ではないが改竄されてはならない機密データ（例えば、認証サーバの公開鍵）がある。いずれのタイプの永続的データでも、敵からの攻撃からデータを守るために、追加の安全性が必要である。さらに、パスワードおよび永続的記憶データの両方に基づく認証プロトコルを使用する場合、一方に対する危険により、他方が攻撃を受けやすくなることがある。例えば、秘密鍵が危険にさらされると、パスワードに対する辞書攻撃が可能となる。この第１分類のプロトコルでのもう１つの問題は、永続的記憶データが鍵の生成および配送を必要とすることである。これは面倒なことがあり、一般に、システムの自由度が低くなる。
【０００８】第２分類は、パスワード単独認証プロトコルと呼ばれる。クライアントに永続的記憶データが不要であるからである。クライアントは、真正なパスワードを提供することができるだけでよい。潜在的に弱いパスワードを用いて強力な安全性および認証を提供するということは矛盾しているように思われるかもしれない。しかし、安全であるように設計されたパスワード単独ユーザ認証および鍵交換プロトコルがいくつか存在する。これらのプロトコルについては、D. Jablon, "Strong Password-Only Authenticated Key Exchange", ACM Computer Communication Review, ACM SIGCOMM, 26(5):5-20, 1996、に記載されている。これらのパスワード単独プロトコルのうち注目に値するものには以下のものがある。
・暗号化された鍵交換（ＥＫＥ：Encrypted Key Exchan

ge) （S. M. Bellovinand M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks", Proceedings of the IEEE Symposium on Research in Security and Privacy, pp.72-84, 1992、を参照）

・拡張ＥＫＥ（Ａ－ＥＫＥ：Augmented-EKE）（S. M. Bellovin and M. Merritt, "Augmented Encrypted Key Exchange: A Password-Based Protocol SecureAgainst Dictionary Attacks and Password File Compromise", Proceedings ofthe First Annual Conference on Computer and Communications Security, 1993, pages 244-250、を参照）

・修正ＥＫＥ（Ｍ－ＥＫＥ：Modified EKE）（M. Steiner, G. Tsudik, and M. Waidner, "Refinement and Extension of Encrypted Key Exchange", ACM Operating System Review, 29:22-30, 1995、を参照）

・単純パスワードＥＫＥ（ＳＰＥＫＥ：Simple Password EKE）およびディフィ・ヘルマンＥＫＥ（ＤＨ－ＥＫＥ：Diffie-Hellman EKE）（D. Jablon, "Strong Password-Only Authenticated Key Exchange", ACM Computer CommunicationReview, ACM SIGCOMM, 26(5):5-20, 1996、を参照）

・安全なリモートパスワードプロトコル（ＳＲＰ：Secure Remote Password Protocol）（T. Wu, "The Secure Remote Password Protocol", Proceedings ofthe 1998 Internet Society Network and Distributed System Security Symposium, pages 97-111, 1998、を参照）

・オープン鍵交換（ＯＫＥ：Open Key Exchange）（Stefan Lucks, "Open KeyExchange: How to Defeat Dictionary Attacks Without Encrypting Public Keys", Security Protocol Workshop, Ecole Normale Superieure, April 7-9, 1997、を参照）

【０００９】

【発明が解決しようとする課題】これらの知られているパスワード単独認証プロトコルの問題点は、これらが安全であることが証明されていないことである。実際、ＥＫＥプロトコルは、いくつかの数論的攻撃を受けやすい可能性がある（S. Patel, "Number Theoretic Attacks on Secure Password Schemes", Proceedings of the IEEE Symposium onResearch in Security and Privacy", pages 236-247, 1997、を参照）。ネットワーク安全性の重要性に鑑み、安全であることが証明可能なパスワード単独相互認証プロトコルが必要とされている。

【００１０】

【課題を解決するための手段】本発明は、公開鍵暗号方式を利用して安全なパスワード単独相互ネットワーク認証プロトコルを提供する。このプロトコルを実現するために用いられる特定の公開鍵暗号方式は、いわゆる使用可能(usable)暗号方式（定義は後述）でなければならない。ネットワークサーバは、公開鍵暗号方式に従って公開鍵・秘密鍵の対を生成し、公開鍵をクライアントに送信する。クライアントは、受信した公開鍵が、公開鍵暗号方式のすべての公開鍵の集合のいわゆるテスト可能スーパーセット（定義は後述）の要素であるかどうかを決定する。この決定が可能であるのは、公開鍵暗号方式が使用可能であることが要求されているからである。公開鍵がテスト可能スーパーセットの要素であるかどうかに関するクライアントによる決定は、クライアントに、サーバが適当な方法で選択した公開鍵を提供したかどうかを決定する方法を提供する。公開鍵がテスト可能スーパーセット内にないことがわかった場合、認証はクライアントによって拒否される。そうでない場合、プロトコルは続行される。

【００１１】本発明の一実施例では、クライアントとサーバはいずれも、認証目的で使用される１つのパスワードを所有する。この実施例では、クライアントは、少なくとも公開鍵およびパスワードの関数としてパラメータｐを生成することによりプロトコルを続行する。公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）が、公開鍵のいわゆるメッセージ空間の要素である場合、クライアントが、公開鍵を用いて公開鍵のメッセージ空間の実質的にランダムな要素を暗号化し、その結果と、F_{PK}（ｐ）との間で、公開鍵メッセージ空間の群演算を実行することによって、プロトコルは続行される。一方、F_{PK}（ｐ）がメッセージ空間の要素でない場合、クライアントは認証を拒否することを決定する。しかし、仮にクライアントがこの時点で拒否をサーバに通知したとすると、サーバはパスワードに関する有用な情報を抽出することができる可能性がある。そのため、クライアントは、認証を拒否することを決定しても、サーバに情報を漏らさないようにプロトコルを続行する。クライアントは、プロトコルの後のほうで、サーバがパスワードに関する有用な情報を得ることができなくなったときに、認証を拒否する。

【００１２】本発明の第２実施例では、サーバにおける安全性の危険に対する保護のために、サーバは、パスワードを所有せず、その代わりにサーバには、パスワードの関数である値が提供され、サーバはそれを記憶する。パスワード自体は、サーバに記憶された値から決定することはできない。

【００１３】本発明の第３および第４の実施例は、使用可能公開鍵暗号方式としてＲＳＡ暗号方式を利用する。これらの実施例によれば、サーバが提供した公開鍵がすべてのＲＳＡ公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するためにＲＳＡ固有テストを行う。さらに、ある値が、ＲＳＡメッセージ空間の要素であるかどうかを決定するためにＲＳＡ固有テストを行う。第３実施例では、サーバが共有パスワードを記憶する。第４実施例では、サーバは、パスワードの関数である値を記憶する。

【００１４】発明者は、本発明による相互認証プロトコルは、基礎となる公開鍵暗号方式と同程度に安全であることを証明した。従って、ＲＳＡ固有の実施例では、発明者は、本発明のプロトコルがＲＳＡ暗号方式と同程度に安全であることを証明したことになる。その証明の概略は後述する。

【００１５】

【発明の実施の形態】暗号は、二者間に安全な通信を提供する周知の技術である。本発明のさまざまな実施例について説明する前に、いくつかの背景知識および基礎的な用語について説明する。

【００１６】まず、暗号方式について説明する。秘密鍵暗号方式では、メッセージｍは、暗号文Ｃを生成するために、暗号化関数Ｅおよび秘密鍵Ｋを用いて暗号化される。これは、$C = E_K(m)$と表される。暗号文Ｃは、秘密鍵Ｋを共有する二者間で安全に送信される。暗号文は、もとのメッセージｍを回復するために、復号関数Ｄおよび秘密鍵Ｋを用いて復号される。これは$m = D_K(C)$と表される。

【００１７】公開鍵暗号方式では、公開鍵（ＰＫ）および秘密鍵（ＳＫ）の対（ＰＫ，ＳＫ）が存在する。公開鍵は秘密ではなく、誰でも公開鍵を用いてメッセージｍを暗号化して$C = E_{PK}(m)$となるような暗号文Ｃを生成することが可能である。暗号文は、秘密鍵を用いてのみ、$m = D_{SK}(C)$と復号可能である。暗号文は、公開鍵を用いては復号できない。公開鍵暗号は当業者に周知である。

【００１８】周知の公開鍵暗号方式の１つにＲＳＡがある。これは、R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signature and Public Key Cryptosystems", Communications of the ACM, vol. 21, 120-126, 1978、に記載されている。ＲＳＡでは、公開鍵は（Ｎ，ｅ）であり、秘密鍵は（Ｎ，ｄ）である。ただし、Ｎは、２つの大きいランダムに選択された素数ｐとｑの積であり（すなわち、Ｎ＝ｐ・ｑ）、ｅは、ｅと（ｐ－１）・（ｑ－１）の最大公約数が１であるような２より大きい任意の数であり、ｄは、e^{-1} ｍｏｄ （ｐ－１）・（ｑ－１）である。暗号化関数は、Ｅ（ｍ）＝m^e ｍｏｄ Ｎであり、復号関数は、Ｄ（Ｃ）＝C^d ｍｏｄ Ｎである。

【００１９】次に、他のいくつかの暗号用語について説明する。非公式的には、集合Ｓから集合Ｔへの関数ｆが一方向性関数であるとは、Ｓ内のすべてのｘに対してｆ（ｘ）を計算するのは容易であるが、Ｔ内のほとんどのｙに対しては、ｆ（ｘ）＝ｙであるようなＳ内のｘを見つけることが計算量的に実現不可能であることである。一方向性関数の一例は、法指数演算である。ｐを大きい素数とし、ｇを、ｍｏｄ ｐの乗法群（すなわち、１，．．．，ｐ－１の範囲の数）の生成元とする。すると、$f(x) = g^x$ ｍｏｄ ｐは一般に一方向性関数であると仮定される。その逆関数（離散対数関数という）は計算が困難である。また、離散対数関数の計算が困難であるこの他の群（例えば、いくつかの楕円曲線群）もある。ディフィ・ヘルマン鍵交換と呼ばれる鍵交換プロトコル（W. Diffie and M. Hellman,"New Directions in Cryptography", IEEE Transactions on Information Theory, vol.22, no.6, 644-654, 1976、を参照）はこの関数に基づいている。具体的には、アリスとボブの二者が次のように秘密鍵を共有する。アリスが、ランダムなｘを選択し、$X = g^x$ ｍｏｄ ｐをボブに送る。一方、ボブは、ランダムなｙを選択し、$Y = g^y$ ｍｏｄ ｐをアリスに送る。秘密鍵を、アリスはY^x ｍｏｄ ｐとして、ボブはX^y ｍｏｄ ｐとして、計算することができる（$Y^x = X^y = g^{xy}$ ｍｏｄ ｐに注意）。また、ディフィ・ヘルマン鍵交換は、離散対数関数の計算が困難であるこの他の群（例えば、いくつかの楕円曲線群）上でも実行可能である。非公式的には、集合Ｓから集合Ｔへの関数ｈがランダムハッシュ関数であるとは、この関数がＳ内の入力ｘで計算されるまでは、ｈの出力がランダムに見える、または、少なくとも予測不能であることをいう。一般にこのようにふるまう既知の関数は、ＳＨＡ－１（FIPS 180-1, "Secure Hash Standard", Federal Information Processing Standards Publication 180-1, 1995、を参照）、およびＲＩＰＥＭＤ－１６０（H. Dobbertin, A. Bosselaers, B. Preneel, "RIPEMD-160: a strengthened version of RIPEMD", in Fast Software Encryption, 3rd Intl. Workshop, 71-82, 1996、を参照）である。

【００２０】一般に、暗号方式は、その安全性のレベルを記述する安全性パラメータを有する。本明細書では、ハッシュ関数の安全性パラメータとしてｋを用い（ただし、$1/2^k$は無視できるほど小さいと仮定する）、公開鍵暗号方式の安全性パラメータとしてｌを用い、特に、ＲＳＡの法Ｎは長さｌビットであると仮定する。

【００２１】本発明の第１実施例による相互認証プロトコルを図１～図２に示す。図の左側に示すステップはサーバによって実行され、図の右側に示すステップはクライアントによって実行される。矢印は、クライアントとサーバの間の通信を表す。このプロトコルによれば、サーバは自分自身をクライアントに対して認証し、クライアントは自分自身をサーバに対して認証する。両側が認証をした後、それぞれ秘密鍵（セッション鍵という）を生成する。この鍵は、その後の安全な通信に使用可能である。

【００２２】プロトコルを開始する前に、クライアントとサーバはある情報を所有していると仮定する。サーバは、使用される特定の公開鍵暗号方式に従って公開鍵・秘密鍵の対（ＰＫ，ＳＫ）を生成する。公開鍵・秘密鍵の対の生成は当業者に周知であり、ここでは説明しない。サーバとクライアントはいずれも、クライアントが

サーバとの認証に使用するパスワードπ（すなわち、共有の秘密）を所有する。パスワードπは、クライアントとサーバの間で事前に設定されなければならず、各クライアント・サーバ対ごとに独立に、あるいは、各クライアント・サーバ対ごとに一意的であるように、選択されるべきである。

【００２３】以下のプロトコルは、サーバおよびクライアントの両方を認証する。従って、サーバおよびクライアントはいずれも真正であるとは仮定されず、サーバまたはクライアントのいずれかが敵である可能性もある。クライアントは、自分自身を認証し、サーバにアクセスしようとしている敵である可能性がある。サーバは、疑いを持っていないクライアントから機密情報を得ようとして別の真正なサーバのふりをしよう（スプーフィング）としている敵である可能性がある。

【００２４】当業者には直ちに明らかなように、サーバおよびクライアントは、コンピュータプログラムコードの制御下で動作するプログラムされたコンピュータとして実現可能である。コンピュータプログラムコードは、コンピュータ可読媒体（例えばメモリ）に記憶され、コードは、コンピュータのプロセッサにより実行される。本明細書および図面が与えられれば、当業者は、ここで説明するプロトコルを実装するために適当なコンピュータプログラムコードを直ちに作成することが可能である。クライアントとサーバはデータネットワークを通じて互いに通信する。このようなネットワーク接続された、プログラムされたコンピュータは当業者に周知であり、ここではこれ以上詳細に説明しない。

【００２５】図１～図２において、プロトコルが開始されると、ステップ１１０で、サーバはｍを生成する。これは、集合Ωのランダムな要素である。Ωは、２つの同値なｍの値を生成する確率が無視できることを保証するほど十分に大きい集合を表す。Ωは、後の鍵交換を可能にするようなものとすることが可能である。ステップ１１２で、サーバは、ｍとＰＫをクライアントに送信する。上記のように、サーバは、プロトコルの開始前に（ＰＫ，ＳＫ）の対を生成していると仮定する。ステップ１１４で、クライアントは、ステップ１１２でサーバから受信したｍが集合Ωに属しているかどうか、および、ＰＫが集合ε'（詳細は後述）に属しているかどうかを決定する。これらのテストのいずれかが偽である場合、クライアントは認証を拒否する。ステップ１１４のテストはクライアントによって実行される。その理由は、真正なサーバのふりをする敵は、クライアントがプロトコルを実行した場合にこの敵がパスワードπに関する何らかの情報を知ることができるようにｍおよびＰＫを選択している可能性があるからである。

【００２６】ここで、ＰＫが集合ε'に属することの意味について説明する。上記のように、プロトコルが正しく動作するとともに機密情報を漏らさないために、公開鍵・秘密鍵の対（ＰＫ，ＳＫ）は、使用される特定の公開鍵暗号方式に従って適当に選択されなければならない。与えられた特定の公開鍵暗号方式に対して、ステップ１１２でサーバから受信したＰＫが、この特定の公開鍵暗号方式を用いて生成されるすべての可能な公開鍵の集合εの要素であるかどうかを決定することができれば理想的である。しかし、適当な時間内にこの決定をすることが可能な公開鍵暗号方式（プロトコルに要求されるすべての性質を有する）は知られていない。そこで、使用可能公開鍵暗号方式およびテスト可能スーパーセットε'を次のように定義する。

【００２７】公開鍵暗号方式が使用可能であるとは、εの次のようなテスト可能スーパーセットε'が存在することである。

１．すべてのＰＫ$\in\varepsilon'$に対して、$s_{PK}=|S_{PK}|$はｋに関して超多項式的である（すなわち、S_{PK}の要素の個数はｋの多項式ではおさえられない）。ただし、S_{PK}は、ＰＫと、すべての可能なメッセージの暗号化とを用いて暗号化可能なすべての可能なメッセージの集合を表す。この集合S_{PK}を、公開鍵（ＰＫ）のメッセージ空間という。

２．任意のＰＫに対して、ＰＫ$\in\varepsilon'$であるかどうかを決定する多項式時間アルゴリズムが存在する。

３．すべてのＰＫ$\in\varepsilon'$に対して、S_{PK}から一様に１つの要素を取り出す期待多項式時間アルゴリズムが存在する。

４．すべてのＰＫ$\in\varepsilon'$に対して、任意の値ａについて、ａ$\in S_{PK}$であるかどうかを決定する多項式時間アルゴリズムが存在する。

５．すべてのＰＫ$\in\varepsilon'$に対して、整数$\eta\geqq 1+k$、$\{0,1\}^{\eta}$を定義域とする多項式時間計算可能な公開鍵空間マッピング関数F_{PK}、およびこの定義域の分割

【数１】

$$X_{PK,1}\cup\cdots\cup X_{PK,s_{PK}}\cup Z_{PK}\cup Z'_{PK}$$

（この分割はＰＫにのみ依存する）が存在して、以下の条件を満たす。

（ａ）s_{PK}は多項式時間で計算可能である。

（ｂ）ｉ$\in\{1,\ldots,s_{PK}\}$に対して、$F_{PK}:X_{PK,i}\rightarrow S_{PK}$は全単射である（すなわち、$F_{PK}$は、各集合$X_{PK,i}$から$S_{PK}$への全単射を含む）。

（ｃ）それぞれのａ$\in S_{PK}$およびｉ$\in\{1,\ldots,s_{PK}\}$に対して、$F_{PK}(x)=a$であるようなｘ$\in X_{PK,i}$を求める多項式時間アルゴリズムが存在する。

（ｄ）与えられたｘ$\in\{0,1\}^{\eta}$に対して、ｘ$\in Z_{PK}$であるかそれともｘ$\in Z'_{PK}$であるかをテストする多項式時間アルゴリズムが存在する。

（ｅ）各ｘ$\in Z_{PK}$に対して、$F_{PK}(x)\in S_{PK}$でない。

（ｆ）$|Z'_{PK}|/2^{\eta}$は、安全性パラメータｋに関して無視できる。

（ｇ）Ｅ∈εである場合、｜Z_{PK}∪Z'_{PK}｜／2^{η}は、安全性パラメータｋに関して無視できる。

【００２８】当業者であれば、この使用可能の定義は、ＰＫを用いて暗号化可能なメッセージの集合が、暗号化されたメッセージの集合に等しくないような暗号方式を含むように直ちに拡張することができる。当業者であれば、ここに記載したプロトコルを、このような暗号方式とともに使用するように直ちに修正することが可能である。こうして、公開鍵ＰＫに関して、ステップ１１４でのテストで、ＰＫが、εのテスト可能スーパーセットε′の要素であるかどうかを決定する。ステップ１１４でのテストが偽である（すなわち、ｍはΩの要素であり、かつ、ＰＫはε′の要素である）場合、認証は続行される。しかし、ステップ１１４でのテストが真である（すなわち、ｍがΩの要素でないか、または、ＰＫがε′の要素でない）場合、認証はクライアントによって拒否される。サーバがＰＫあるいはｍを不適当な方法で選択したからである。

【００２９】ステップ１１６で、クライアントは、パラメータμを、集合Ωのランダムな要素として設定する。ステップ１１８で、クライアントは、パラメータａを、メッセージ空間S_{PK}のランダムな要素として設定する。ステップ１２０で、クライアントは、パラメータ（ＰＫ，ｍ，μ，π）のランダムハッシュ関数Ｈとしてパラメータｐを計算する。ハッシュ関数Ｈは、十分なビット数（少なくともη）を出力する上記のような任意のランダムハッシュ関数とすることが可能である。

【００３０】ステップ１２２で、公開鍵空間マッピング関数F_{PK}をｐに作用させた結果F_{PK}（ｐ）がメッセージ空間S_{PK}の要素であるかどうかを決定する。F_{PK}（ｐ）がS_{PK}の要素でない場合、認証は拒否される。しかし、ステップ１２２で、F_{PK}（ｐ）∈S_{PK}でないと決定された場合、この時点で認証を終了するのは好ましくない。その理由は、クライアントがこの時点で認証を終了した場合、敵のサーバが、パスワードπについての何らかの知識を得る可能性があるからである。従って、クライアントは、認証を拒否すると決定した場合でも、サーバとのプロトコルを続行することが好ましい。そこで、ステップ１２２でのテストが真である場合、クライアントはｑ＝ａと設定する。ｑ＝ａと設定する（ａは、ステップ１１８で、メッセージ空間の実質的にランダムな要素として選択された）ことによって、敵のサーバは、パスワードπに関する情報を得ることがなくなる。ステップ１２２で、F_{PK}（ｐ）∈S_{PK}と決定された場合、クライアントは、ｑ＝E_{PK}（ａ）○F_{PK}（ｐ）を計算することによって認証プロトコルを進める。すなわち、ａは公開鍵を用いて暗号化され、F_{PK}（ｐ）が、公開鍵メッセージ空間の群演算を用いて、暗号化結果に作用する。ステップ１２４で、μ、ｑがサーバに送られる。

【００３１】ステップ１２６で、サーバは、μ∈Ωかつｑ∈S_{PK}であるかどうかを決定する。μ∈Ωでないか、または、ｑ∈S_{PK}でない場合、サーバは認証を拒否する。そうでない場合、ステップ１２８で、サーバは、パラメータ（ＰＫ，ｍ，μ，π）のランダムハッシュ関数Ｈとしてパラメータｐ′を計算する。このステップ１２８は、ステップ１２０について既に説明したのとに実行される。ステップ１３０で、サーバは、F_{PK}（ｐ′）がメッセージ空間S_{PK}の要素であるかどうかを決定し、S_{PK}の要素でない場合、サーバは認証を拒否する。F_{PK}（ｐ′）がメッセージ空間S_{PK}の要素である場合、認証は続行される。ステップ１３２で、サーバは、秘密鍵ＳＫを用いてｑ／F_{PK}（ｐ′）を復号することによってａ′を計算する（ここで／は、公開鍵メッセージ空間の群演算の逆を表す）。ステップ１３４で、サーバは、ランダムハッシュ関数ｈをａ′に作用させたものとしてｒを計算する。ステップ１３６で、サーバはｒをクライアントに送る。

【００３２】ステップ１３８で、クライアントは、F_{PK}（ｐ）∈S_{PK}かつｒ＝ｈ（ａ）であるかどうかを決定する。これらの条件が両方とも真である場合に限り、クライアントはサーバを真正であるとして受け入れる。なお、F_{PK}（ｐ）∈S_{PK}でない場合には、クライアントは既にステップ１２２でサーバを受け入れないと決定しているが、敵がステップ１２２での認証の拒否から情報を得ることができないように認証プロトコルを続行していたことを想起すべきである。ｒ＝ｈ（ａ）のテストは、サーバが正しいパスワードπを所有していたかどうかをテストする。そのため、ステップ１３８で、F_{PK}（ｐ）∈S_{PK}でないか、または、ｒ≠ｈ（ａ）であるためにクライアントが認証を拒否しても、サーバは、いかなる理由で認証が拒否されたかを決定することができない。ステップ１３８で、クライアントが、サーバを受け入れると決定した場合、ステップ１４０で、クライアントは、ｔ＝ｈ′（ａ）を計算する。ただし、ｈ′はランダムハッシュ関数である。ステップ１４２で、クライアントはｔをサーバに送る。

【００３３】ステップ１４６で、サーバは、ｔ＝ｈ′（ａ′）であるかどうかを決定する。ｔ＝ｈ′（ａ′）である場合、サーバは認証を受け入れる。そうでない場合、サーバは認証を拒否する。クライアントおよびサーバがいずれも認証を受け入れた場合、ステップ１４４で、クライアントはセッション鍵を計算し、ステップ１４８で、サーバがセッション鍵を計算する。セッション鍵は、共有秘密鍵として作用し、クライアントとサーバの間のその後の安全な通信のために使用される。このような方法での秘密鍵の使用は、公開鍵暗号の継続使用よりも、その後の安全な通信にとって効率的である。一実施例では、セッション鍵Ｋは、サーバおよびクライアントの両方で、ａのランダムハッシュ関数ｈ″として、Ｋ＝ｈ″（ａ）と計算される。代替実施例では、セッショ

ン鍵Ｋは、サーバおよびクライアントの両方で、ディフィ・ヘルマンプロトコルを用いて、ｍおよびμをディフィ・ヘルマンパラメータに選択して、計算される。当業者には明らかなように、セッション鍵を計算するために、さまざまな代替法を使用可能である。

【００３４】このようにして、図１～図２について説明したプロトコルは、上記で定義したように使用可能な公開鍵暗号方式を用いて、クライアントとサーバの相互認証を実現する。図１～図２について説明したプロトコルは、サーバがパスワードπを所有し記憶していることを仮定している。このようなプロトコルの１つの潜在的な問題点は、サーバ記憶領域の安全性が危険にさらされることにより、敵が、クライアントのパスワードを取得することが可能になることである。このような事態に対する保護のために、本発明の第２実施例では、サーバはパスワードπを所有せず、代わりに、パスワードπとｓａｌｔ値の関数である値Ｘを記憶する。ｓａｌｔ値は、敵が各ｓａｌｔ値ごとに別の辞書攻撃を実行するよう強制することによって、複数のパスワードに対する同時の辞書攻撃を妨げるために使用される公知の値である。値Ｘは、クライアントによってサーバに供給され、従って、サーバは、Ｘを知るのみであり、Ｘの知識からπを決定することはできない。クライアントは、ＸをＸ＝ｇ[x]として計算する。ただし、ｇは離散対数の計算が困難なある群の生成元であり、ｘ＝Ｈ′（π，ｓａｌｔ）であり、Ｈ′は一方向性ランダムハッシュ関数を表す。図１および図２に示した第１実施例について上記で説明したのと同様に、サーバは、使用される特定の公開鍵暗号方式に従って公開鍵・秘密鍵の対（ＰＫ，ＳＫ）を生成する。

【００３５】以下、本発明の第２実施例によるプロトコルについて、図３～図４とともに説明する。ステップ２０５で、クライアントは、クライアントのユーザ名をサーバに送ることによってプロトコルを開始する。ステップ２１０で、サーバは、Ωのランダムな要素としてｍを生成する。ステップ２１２で、サーバは、ステップ２０５で受信したユーザ名に対応するＸおよびｓａｌｔを記憶領域から取得する。ステップ２１４で、サーバは、ｍ、ＰＫおよびｓａｌｔをクライアントに送る。ステップ２１６で、クライアントは、ステップ２１４でサーバから受信したｍが集合Ωに属しているかどうか、および、ＰＫがテスト可能スーパーセットε′に属しているかどうかを決定する。これらのテストのいずれかが偽である場合、クライアントは認証を拒否する。そうでない場合、認証は続行され、ステップ２１８で、クライアントは、ステップ２１４でサーバから受信したｓａｌｔを用いてｘ＝Ｈ′（π，ｓａｌｔ）を計算する。ただし、Ｈ′はランダムハッシュ関数である。ステップ２２０で、クライアントは、パラメータμを、集合Ωのランダムな要素として設定し、ステップ２２２で、クライアントは、パラメータａを、メッセージ空間S_{PK}のランダムな要素として設定する。ステップ２２４で、クライアントは、パラメータ（ＰＫ，ｍ，μ，g^x）のランダムハッシュ関数Ｈとしてパラメータｐを計算する。このステップ２２４は、第１実施例のステップ１２０と同様であるが、この第２実施例では、ｐを決定する際のパラメータの１つとしてパスワードπを使用する代わりに、パスワードπの関数、すなわちg^xを、パラメータの１つとして使用する。その後、ステップ２２６、２２８、および２３０は、上記のステップ１２２、１２４、および１２６で説明したのと同様に実行される。

【００３６】ステップ２３２で、サーバは、パラメータ（ＰＫ，ｍ，μ，Ｘ）のランダムハッシュ関数Ｈとしてパラメータｐ′を計算する。このステップは、第１実施例のステップ１２８と同様であるが、ステップ２３２では、サーバはπを知らないため、代わりに、ハッシュ関数のパラメータとしてＸを使用する。ステップ２３４、２３６、および２３８は、上記のステップ１３０、１３２、および１３４で説明したのと同様に実行される。ステップ２４０で、集合Ｗのランダムな要素としてγを選択する。ただし、Ｗは、相異なるg^γの値からなる十分に大きな集合を生じるｇの可能な指数の集合を表す。ステップ２４２で、ｙをg^γとおく。ステップ２４４で、サーバは、ｒおよびｙをクライアントに送る。ステップ２４６は、上記のステップ１１３８で説明したのと同様に実行される。ステップ２４８で、クライアントはｔ＝ｈ′（ａ，y^x）を計算し、ステップ２５０で、クライアントはｔをサーバに送る。

【００３７】ステップ２５４で、サーバは、ｔ＝ｈ′（ａ′，X^γ）であるかどうかを決定する。ｔ＝ｈ′（ａ′，X^γ）である場合、サーバは認証を受け入れる。そうでない場合、サーバは認証を拒否する。クライアントおよびサーバがいずれも認証を受け入れた場合、ステップ２５２で、クライアントはセッション鍵を計算し、ステップ２５６で、サーバがセッション鍵を計算する。

【００３８】図１～図２および図３～図４を参照して上記でそれぞれ説明した本発明の第１および第２実施例は、上記のように使用可能であるという要件を満たす公開鍵暗号方式とともに用いられる認証プロトコルである。このような使用可能な公開鍵暗号方式の１つは、パラメータを以下で説明するいくつかの制約に従って選択したＲＳＡ公開鍵暗号方式である。以下、本発明の第３および第４の実施例について説明する。第３実施例は、ＲＳＡを、サーバにパスワードπが記憶された公開鍵暗号方式として利用する。第４実施例は、値Ｘがサーバに記憶されたＲＳＡ公開鍵暗号方式を利用する。ただし、Ｘは、パスワードπとｓａｌｔ値の関数である。このように、第３および第４の実施例は、第１および第２実施例にそれぞれ対応するＲＳＡ固有の実施例である。

【００３９】以下、本発明の第３実施例について、図５～図６とともに説明する。ＲＳＡ公開鍵暗号方式では、公開鍵ＰＫは２個のパラメータ（Ｎ，ｅ）から構成され、秘密鍵ＳＫは２個のパラメータ（Ｎ，ｄ）から構成される。使用可能な形のＲＳＡ公開鍵暗号方式では、Ｎは大きく、ｅは、（Ｎ，ｅ）を知っている者がＮの任意の素因数ｒに対してｅと（ｒ－１）の最大公約数が１であることを容易にテストすることができるという性質を有することが保証されるように、公開鍵ＰＫ（Ｎ，ｅ）は選択される。これを実行するいくつかの適当な方法について以下で説明する。サーバは、プロトコルの開始前に、適当な（ＰＫ，ＳＫ）の対を生成していると仮定する。ステップ３０２で、サーバは、Ωのランダムな要素としてｍを生成する。ステップ３０４で、サーバは、ｍ、Ｎ、およびｅをクライアントに送信する。図１のステップ１１４について説明したように、クライアントは、次に、サーバから受信したｍおよび公開鍵ＰＫが、敵がパスワードπに関する情報を知り得るような方法でこれらの値を選択することに対する保護に適当であるように選択される。次に、ステップ３０６で、クライアントは、ステップ３０４でサーバから受信したｍが集合Ωに属しているかどうか、および、ＰＫがテスト可能スーパーセットε′（上記で定義）に属しているかどうかを決定する。ＲＳＡによる実装では、ＰＫがテスト可能スーパーセットε′に属しているかどうかを決定する１つの方法は、ＮおよびｅがＮの次の条件を満たすかどうかを決定することである。

$N \in [2^{l-2}, 2^{l}]$

$e \in [2^{l}, 2^{l+1}]$

ｅは素数である

ＮおよびｅがＮの上記の条件をすべて満たす場合、ＰＫ∈ε′である。これらの条件は、Ｎが十分に大きいか（Ｎが2^{l-2}～2^{l}の範囲内にあるかどうかを決定することにより）、および、ｅがＮより大きいか（ｅが2^{l}～2^{l+1}の範囲内にあるかどうかを決定することにより）のテストを含む。図５のステップ３０６に示すいずれかの条件が偽である場合、認証は拒否される。そうでない場合、プロトコルはステップ３０８に進む。なお、ＲＳＡ実施例では、ＰＫ∈ε′であるかどうかを決定するもう１つの代替テスト法がある。この代替テストは、次の条件が満たされるかどうかを決定することである。

$e \geqq \sqrt{N}$

Ｎ　ｍｏｄ　ｅはＮで割り切れない

ｅは素数である

Ｎおよびｅがこれらの条件をすべて満たす場合、ＰＫ∈ε′である。

【００４０】なお、ＲＳＡ固有の実施例では、ＰＫ∈ε′であるかどうかを決定するために使用可能な他のテスト法もある。例えば、ｅは素数であるかどうかをテストする代わりに、ｅを公知の固定値とし、ｅがこの固定値に等しいことを確認するテストを行うことが可能である。当業者であれば、ＲＳＡ固有の実施例でＰＫ∈ε′であるかどうかを決定するための他のテストを実装することが可能である。

【００４１】ステップ３０８、３１０、３１２はそれぞれ、上記の図１のステップ１１６、１１８、１２０で説明したのと同様に実行される。なお、ステップ１２０（図１）における計算は、ステップ３１２（図５）に示したものと同じであるが、図１の一般的なＰＫは、図５～図６のＲＳＡ固有の実装ではＮおよびｅで置き換えられる。次のテストは、F_{PK}（ｐ）が、ＲＳＡ公開鍵のメッセージ空間の要素である（すなわち、F_{PK}（ｐ）∈S_{PK}）かどうかを決定する。ＲＳＡ固有の実施例では、これは、ｐとＮの最大公約数（ｇｃｄ）が１に等しいかどうかを決定することによって実行される。ｇｃｄ（ｐ，Ｎ）＝１である場合、F_{PK}（ｐ）∈S_{PK}であり、ステップ３１４で、クライアントは、$q = (p \cdot a^{e}) \bmod N$を計算する。ｇｃｄ（ｐ，Ｎ）≠１である場合、$F_{PK}$（ｐ）∈$S_{PK}$でないため、認証は拒否される。しかし、図１のステップ１２２に関して既に述べたように、この時点で認証を終了するのは好ましくない。その理由は、そうすると、敵のサーバが、パスワードπについての何らかの知識を得る可能性があるからである。従って、クライアントは、認証を拒否すると決定した場合でも、サーバとのプロトコルを続行することが好ましい。そこで、ステップ３１４でのテストが真である場合、クライアントは、ｑ＝ａとおくことにより、ｑをメッセージ空間の実質的にランダムな要素に設定する。ステップ３１６で、クライアントは、μ、ｑをサーバに送信する。

【００４２】ステップ３１８で、サーバは、μ∈Ωかつｑ∈S_{PK}であるかどうかを決定する。ｑ∈S_{PK}である（すなわち、ｑがＲＳＡ公開鍵のメッセージ空間の要素である）かどうかのテストは、ｇｃｄ（ｑ，Ｎ）＝１であるかどうかをテストすることにより行われる。μ∈Ωでないか、または、ｇｃｄ（ｑ，Ｎ）≠１である場合、サーバは認証を拒否する。そうでない場合、ステップ３２０で、サーバは、パラメータ（Ｎ，ｅ，ｍ，μ，π）のランダムハッシュ関数Ｈとしてパラメータｐ′を計算する。ステップ３２２で、サーバは、ｇｃｄ（ｐ′，Ｎ）＝１であるかどうかを決定することにより、F_{PK}（ｐ′）がＲＳＡ公開鍵のメッセージ空間の要素であるかどうかを決定する。ｇｃｄ（ｐ′，Ｎ）≠１である場合、F_{PK}（ｐ′）はＲＳＡ公開鍵のメッセージ空間の要素ではなく、サーバは認証を拒否する。ｇｃｄ（ｐ′，Ｎ）＝１である場合、F_{PK}（ｐ′）はＲＳＡ公開鍵のメッセージ空間の要素であり、認証は続行される。ステップ３２４で、サーバは、（ｑ／ｐ′）に対してＲＳＡ復号を実行する。ステップ３２６～３４０は、それぞれ図２のステップ１３４～１４８と同様に実行さ

れる。なお、この第３実施例はＲＳＡ固有であるが、F_{PK}（ｐ）∈S_{PK}であるかどうかを決定するステップ３３０（図２のステップ１３８に対応する）におけるテストは、ｇｃｄ（ｐ，Ｎ）＝１であるかどうかを決定することにより実行される。

【００４３】本発明の第４実施例について図７～図８を参照して説明する。サーバは、パスワードπおよびｓａｌｔ値の関数である値Ｘを記憶する。これは第２実施例と同様である。しかし、第２実施例は任意の使用可能な公開鍵暗号方式を用いたプロトコルであった。第４実施例は、公開鍵暗号方式としてＲＳＡを使用する。図７において、ステップ４０２で、クライアントは、ユーザ名をサーバに送ることによってプロトコルを開始する。ステップ４０４で、サーバは、Ωのランダムな要素としてｍを生成する。ステップ４０６で、サーバは、ステップ４０２で受信したユーザ名に対応するＸおよびｓａｌｔを記憶領域から取得する。ステップ４０８で、サーバは、ｍ、Ｎ、ｅ、およびｓａｌｔをクライアントに送る。ステップ４１０で、クライアントは、ステップ４０８でサーバから受信したｍが集合Ωに属しているかどうか、および、公開鍵Ｎ、ｅがテスト可能スーパーセットε′に属しているかどうかを決定する。このＲＳＡ固有の実施例では、公開鍵がテスト可能スーパーセットε′に属しているかどうかのステップ４１０におけるテストは、上記の図５とともに説明したステップ３０６におけるテスト（または、上記のステップ３０６とともに説明した代替テスト）と同じである。ステップ４１２、４１４、および４１６で、クライアントは、それぞれ上記の図３のステップ２１８、２２０、および２２２とともに説明したようにして、パラメータｘ、μ、ａを生成する。ステップ４１８で、クライアントは、パラメータ（Ｎ，ｅ，ｍ，μ，g^{x}）のランダムハッシュ関数Ｈとしてパラメータｐを計算する。次に、ステップ４２０で、クライアントは、ｇｃｄ（ｐ，Ｎ）＝１であるかどうかを決定することによって、F_{PK}（ｐ）∈S_{PK}であるかどうかを決定し、上記の図５のステップ３１４とともに説明したのと同様に適当にパラメータｑを生成する。その後、ステップ４２２で、クライアントは、μ、ｑをサーバに送信する。

【００４４】ステップ４２４で、サーバは、μ∈Ωかつｑ∈S_{PK}であるかどうかを決定する。ｑ∈S_{PK}であるかどうかのテストは、ｇｃｄ（ｑ，Ｎ）＝１であるかどうかをテストすることにより行われる。μ∈Ωでないか、または、ｇｃｄ（ｑ，Ｎ）≠１である場合、サーバは認証を拒否する。そうでない場合、ステップ４２６で、サーバは、パラメータ（Ｎ，ｅ，ｍ，μ，Ｘ）のランダムハッシュ関数Ｈとしてパラメータｐ′を計算する。その後、ステップ４２８、４３０、４３２は、それぞれ図５～図６のステップ３２２、３２４、３２６と同様に実行され、ステップ４３４、４３６は、それぞれ図４のステップ２４０、２４２と同様に実行される。

【００４５】ステップ４３８で、サーバは、ｒおよびｙをクライアントに送る。ステップ４４０～４５０は、それぞれ図４のステップ２４６～２５６と同様に実行される。ステップ４４０のF_{PK}（ｐ）∈S_{PK}であるかどうかのテストは、ｇｃｄ（ｐ，Ｎ）＝１であるかどうかをテストすることにより、ＲＳＡ固有の方法で実行される。

【００４６】本発明の発明者は、本発明による相互認証プロトコルは、基礎となる公開鍵暗号方式と同程度に安全であることを証明した。従って、ＲＳＡ固有の実施例では、発明者は、本発明のプロトコルがＲＳＡ暗号方式と同程度に安全であることを証明したことになる。その証明の概略を述べる。

【００４７】本発明が安全な相互認証プロトコルであることを証明するため、プロトコルの安全性の議論を、使用される暗号化関数の安全性の議論に帰着させる。具体的には、入力として暗号化関数および暗号文をとるシミュレータを考え、以下の事象のうちの１つが起こるように、敵に対してランダムに選択されたパスワードでプロトコルをシミュレートする。

１．いくつかのランダムに選択された値が衝突する（すなわち、等しくなる）ことがまれに起こる。これは、非常に低い確率βで起こることが示される。

２．シミュレータは、無視できない確率で、暗号文に対する復号を導出できるという事象がある。

３．上記以外。この場合、敵がプロトコルを破る確率は高々ν／ｄであることを直接に証明する。ただし、νはアクティブな「スプーフィング」攻撃の数であり、ｄは可能なパスワードの数である。

【００４８】ここで、敵が、無視できないεに対して、確率ν／ｄ＋εでプロトコルを破るとする。（非公式的には、これは、敵が、単にパスワードを推測してそれぞれでログインしようとするより実質的に高い確率でプロトコルを破ることができることを意味する。これは不可能であることをわれわれは主張する。）この敵を用いると、入力として暗号化関数および暗号文をとり、無視できない確率で暗号文を復号するアルゴリズムＡを構成することができることになる。Ａは、シミュレータに対する敵を実行する。E_1、E_2およびE_3を上記の３つの事象とし、Ｂを、敵がプロトコルを破る事象とする。上記の議論から、次のことがわかる。

【数２】

$$\frac{v}{d}+\varepsilon \leq \Pr(B)$$
$$\leq \Pr(E_1)+\Pr(B\wedge\overline{E}_1)$$
$$=\Pr(E_1)+\Pr(B\wedge E_2\wedge\overline{E}_1)+\Pr(B\wedge\overline{E}_2\wedge\overline{E}_1)$$
$$\leq \Pr(E_1)+\Pr(E_2)+\Pr(B\wedge E_3)$$
$$=\Pr(E_1)+\Pr(E_2)+\Pr(B|E_3)\cdot\Pr(E_3)$$
$$\leq \Pr(E_1)+\Pr(E_2)+\Pr(B|E_3)$$
$$\leq \beta+\Pr(E_2)+(\frac{v}{d})$$

従って、Ｐｒ（E_2）≧ε－βであり、これは無視できない。従って、Ａは、無視できない確率で暗号文を復号することができるが、これは、暗号化関数の安全性と矛盾する。

【００４９】

【発明の効果】以上述べたごとく、本発明によれば、安全であることが証明可能なパスワード単独相互認証プロトコルが得られる。

【図面の簡単な説明】

【図１】サーバがパスワードを記憶する認証プロトコルの実施例を示す図である。

【図２】サーバがパスワードを記憶する認証プロトコルの実施例を示す図である。

【図３】サーバが、パスワードの関数である値を記憶する認証プロトコルの実施例を示す図である。

【図４】サーバが、パスワードの関数である値を記憶する認証プロトコルの実施例を示す図である。

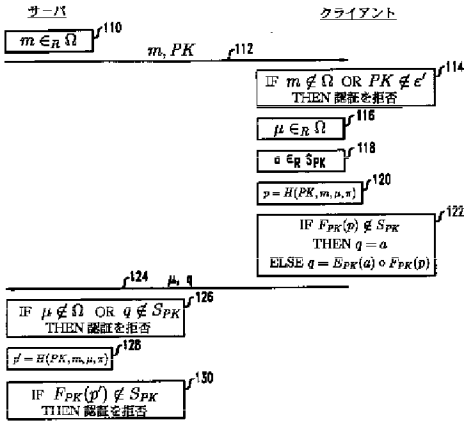
【図５】サーバがパスワードを記憶する認証プロトコルのＲＳＡ固有の実施例を示す図である。

【図６】サーバがパスワードを記憶する認証プロトコルのＲＳＡ固有の実施例を示す図である。

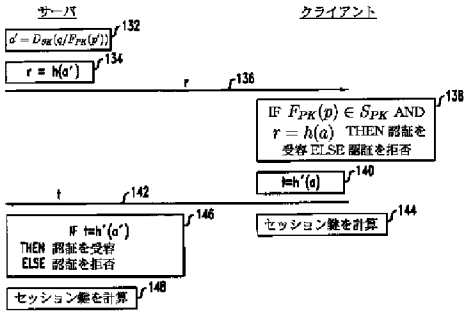
【図７】サーバが、パスワードの関数である値を記憶する認証プロトコルのＲＳＡ固有の実施例を示す図である。

【図８】サーバが、パスワードの関数である値を記憶する認証プロトコルのＲＳＡ固有の実施例を示す図である。

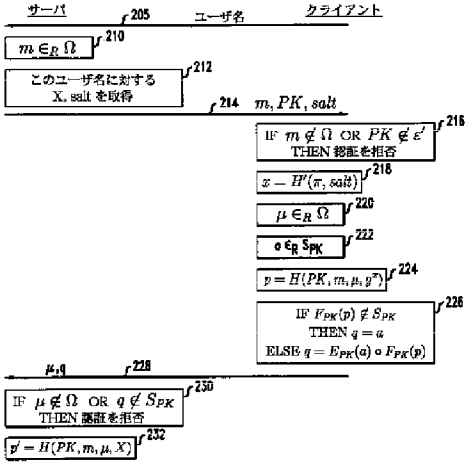
【図１】



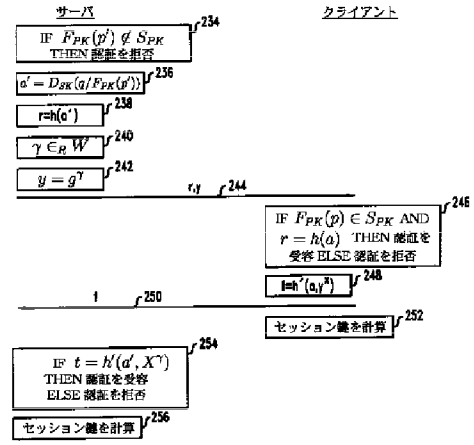
【図２】



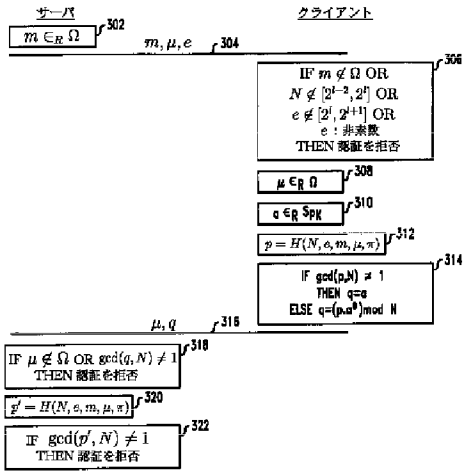
【図３】



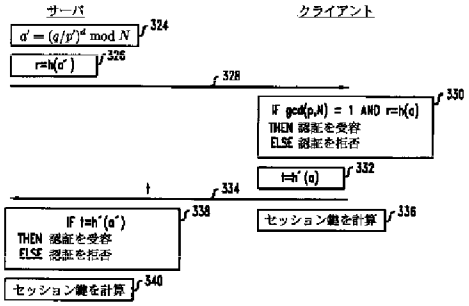
【図４】



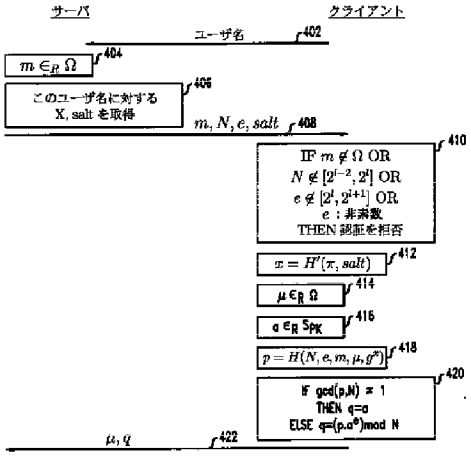
【図５】



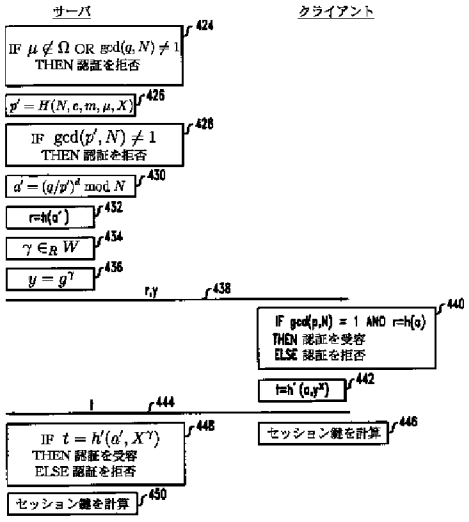
【図６】



【図７】



【図８】



フロントページの続き

(71)出願人　596077259
600　Ｍｏｕｎｔａｉｎ　Ａｖｅｎｕｅ，Ｍｕｒｒａｙ　Ｈｉｌｌ，　Ｎｅｗ　Ｊｅｒｓｅｙ　07974－0636Ｕ．Ｓ．Ａ．

(72)発明者　フィリップ　ダグラス　マッケンジー
アメリカ合衆国、07040　ニュージャージー、メイプルウッド、カーレトン　コート　11

(72)発明者　ラム　スワミナサン
アメリカ合衆国、07974　ニュージャージー、ニュー　プロビデンス、ガレス　ドライブ　＃３　72